# Isomap:
# A Non-linear Dimensionality Reduction Technique

Ahmed BADI
ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

January 2026

**Abstract**

Isomap (Isometric Mapping) is a nonlinear dimensionality reduction algorithm that uncovers low-dimensional manifolds embedded in high-dimensional spaces. It builds on classical Multidimensional Scaling (MDS) by replacing straight-line Euclidean distances with geodesic distances approximated on a neighborhood graph. This allows Isomap to preserve the intrinsic geometry of non-linear manifolds, such as the famous Swiss roll, where Euclidean distances are misleading. In this article, we introduce the intuition behind Isomap, present the algorithm step by step, derive its main mathematical components, and discuss practical aspects such as neighborhood selection and computational cost. We also compare Isomap with PCA, LLE, t-SNE and UMAP, and highlight typical applications and limitations.

**Keywords:** Isomap, Nonlinear Dimensionality Reduction, Geodesic Distance, Manifold Learning, MDS.

## 1 Introduction

Many high-dimensional datasets actually lie on lower-dimensional, curved manifolds. For example, images of an object under varying poses or lighting conditions form a nonlinear manifold in pixel space. Linear methods such as PCA fail to capture this curved structure because they preserve straight-line (Euclidean) distances. [1], [2]

Isomap, introduced by Tenenbaum, de Silva and Langford (2000), combines ideas from manifold learning and classical MDS. The key idea is to approximate *geodesic distances* along the manifold using shortest paths on a neighborhood graph, and then find a low-dimensional embedding that preserves those distances as well as possible. [3], [4]

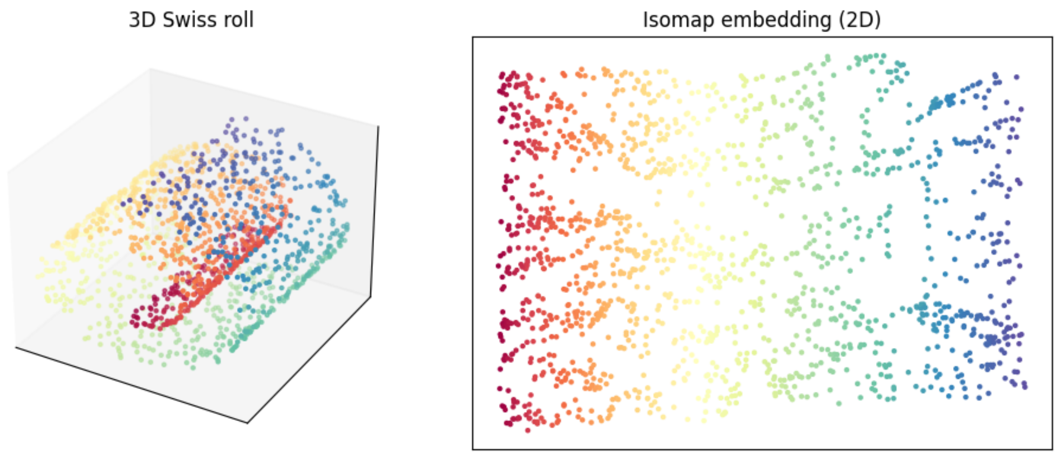


Figure 1: Isomap unfolding a 3D Swiss roll into a 2D manifold by preserving geodesic distances.

Isomap is particularly useful when global manifold structure matters, in contrast to methods like t-SNE that focus mainly on local clusters. [5], [6]

## 2 From MDS to Isomap

### 2.1 Classical MDS

Classical Multidimensional Scaling (MDS) takes as input a matrix of pairwise distances ($D_{ij}$) and finds a configuration of points $\{\mathbf{y}_i\}$ in low dimension such that their Euclidean distances approximate $D_{ij}$. [1], [7]

Given squared distances $D_{ij}^2$, MDS constructs a centered Gram matrix

$$B = -\frac{1}{2} J D^{(2)} J,$$

where $D^{(2)}$ is the matrix with entries $D_{ij}^2$, and

$$J = I - \frac{1}{N} \mathbf{1}\mathbf{1}^\top$$

is the centering matrix, with $N$ the number of points.

MDS then computes the eigen-decomposition

$$B = Q \Lambda Q^\top,$$

where $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_N)$ with $\lambda_1 \geq \lambda_2 \geq \dots$, and $Q$ contains the corresponding eigenvectors. The low-dimensional embedding in $\mathbb{R}^d$ is given by

$$Y = [\sqrt{\lambda_1}\mathbf{q}_1, \dots, \sqrt{\lambda_d}\mathbf{q}_d],$$

where $\mathbf{q}_k$ is the $k$-th eigenvector. [7]

Classical MDS assumes the distance matrix is Euclidean. When data lie on a curved manifold, Euclidean distances between distant points do not reflect manifold distances.

### 2.2 Isomap: Replace Euclidean with Geodesic Distances

Isomap extends MDS by replacing Euclidean distances with estimates of *geodesic distances* along the manifold. [1], [4]

Intuitively:

- Short distances between nearby points are approximated by Euclidean distances.
- Long distances between far points are approximated by the sum of short distances along a path that follows the manifold.

By feeding this geodesic distance matrix into MDS, Isomap obtains an embedding that preserves manifold geometry rather than raw Euclidean structure. [8], [9]

## 3 Algorithm Description

Given data points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^D$, target dimension $d$, and neighborhood parameter (either number of neighbors $k$ or radius $\epsilon$), the Isomap algorithm proceeds in three main steps. [2], [10], [11]

## 3.1 Step 1: Construct Neighborhood Graph

Compute pairwise Euclidean distances

$$\delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|,$$

for all pairs $i, j$. Then construct a weighted graph $G = (V, E)$ where:

- Each node represents a data point.
- Edges connect either:
  - the $k$ nearest neighbors of each point (k-NN Isomap), or
  - all points within radius $\epsilon$ (epsilon-Isomap).
- Edge weights are Euclidean distances $\delta_{ij}$ for connected pairs; non-neighbors are assigned infinite or very large distances.

This neighborhood graph is meant to approximate local manifold structure. [10], [12]

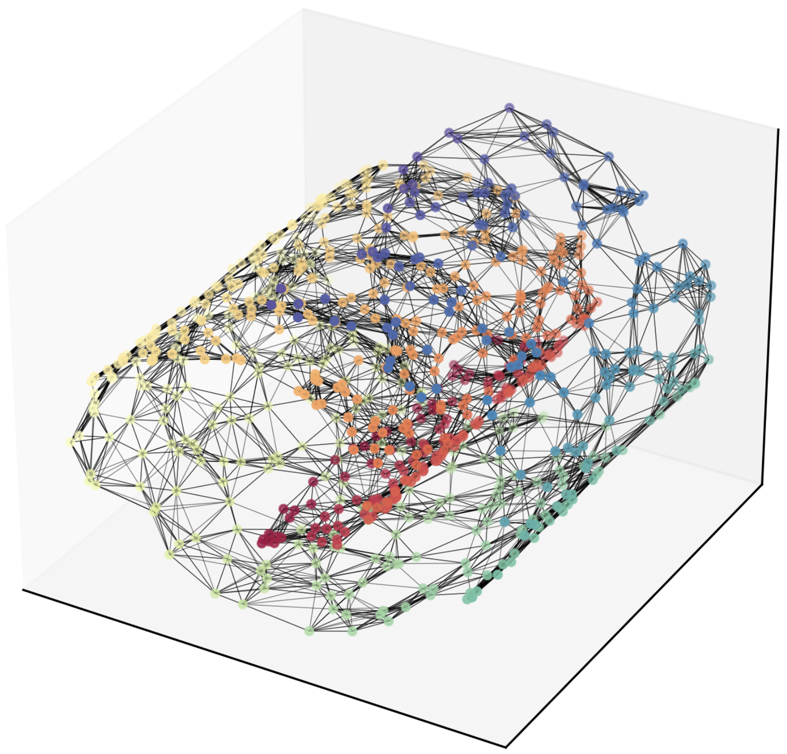


Figure 2: Neighborhood graph used by Isomap: edges connect nearby points along the manifold.

## 3.2 Step 2: Approximate Geodesic Distances

To approximate geodesic distances between all pairs of points, Isomap computes shortest path distances in the neighborhood graph. [8]

Let $G(i, j)$ denote the geodesic distance estimate between points $i$ and $j$. It is defined as the length of the shortest path connecting $i$ and $j$ in the graph (sum of edge weights along the path). This can be computed using:

- Floyd–Warshall algorithm (all-pairs shortest paths), or
- Dijkstra's algorithm from each node.

Mathematically, if $\mathcal{P}_{ij}$ is the set of all paths between $i$ and $j$, and $\ell(P)$ is the length of path $P$, then

$$G(i,j) = \min_{P \in \mathcal{P}_{ij}} \ell(P).$$

These geodesic distances better reflect the manifold's intrinsic geometry than raw Euclidean distances. [8], [12]

### 3.3 Step 3: Apply Classical MDS

Form the matrix $D$ of geodesic distances:

$$D_{ij} = G(i,j).$$

Apply classical MDS to $D$:

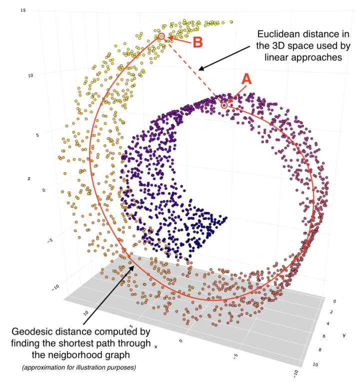
1. Compute squared distances $D_{ij}^2$ and the centered Gram matrix

$$B = -\frac{1}{2} J D^{(2)} J.$$

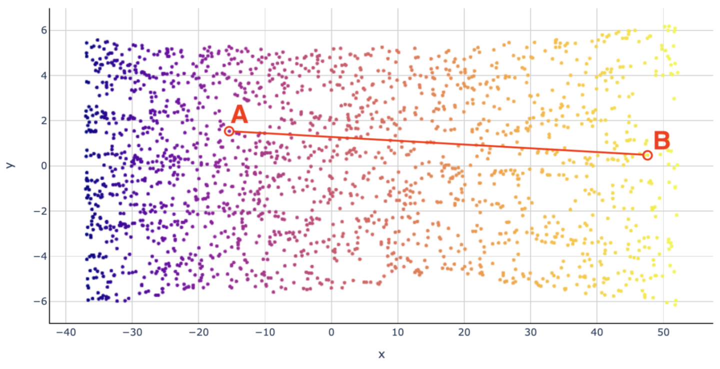
2. Compute eigen-decomposition $B = Q \Lambda Q^\top$.
3. Take the top $d$ eigenvalues and eigenvectors, and form the embedding

$$Y = [\sqrt{\lambda_1}\mathbf{q}_1, \ldots, \sqrt{\lambda_d}\mathbf{q}_d].$$

The rows of $Y \in \mathbb{R}^{N \times d}$ are the low-dimensional coordinates of the data that approximately preserve geodesic distances. [1], [3]



(a) Isomap embedding — view 1



(b) Isomap embedding — view 2

Figure 3: Isomap 2D embeddings: geodesic distances on the manifold are preserved in the 2D space.

Source images adapted from: Towards Data Science — Isomap Embedding article

## 4 Parameter Choices and Complexity

### 4.1 Neighborhood Size

The neighborhood size (either $k$ or $\epsilon$) is critical: [7], [10]

- Too small: the graph may become disconnected; geodesic distances between components are undefined.

- Too large: the graph may include edges that “shortcut” the manifold, turning geodesic distances into approximate Euclidean ones and destroying nonlinear structure.

In practice, different values of $k$ are tried, and the one that yields a connected graph and reasonable embeddings is chosen.

### 4.2 Computational Complexity

Main cost components: [9], [13]

- Pairwise distances and neighbor search: typically $O(N^2)$ for naive implementations.
- Shortest paths: Floyd–Warshall is $O(N^3)$; using Dijkstra from each node on a sparse graph can be $O(N^2 \log N)$.
- Eigen-decomposition of the $N \times N$ Gram matrix: typically $O(N^3)$, though only the top $d$ eigenvectors are needed and can be computed with iterative methods.

Thus Isomap is more expensive than linear methods such as PCA and can struggle with very large datasets, though approximate variants and landmark-based approaches exist. [14], [15]

## 5 Advantages and Limitations

### 5.1 Advantages

- **Captures non-linear structure:** Isomap can unfold curved manifolds like the Swiss roll that PCA cannot. [2]
- **Preserves global geometry:** By preserving geodesic distances, Isomap maintains global manifold structure, not just local neighborhoods. [5], [6]
- **Conceptually simple:** It is a clean extension of classical MDS with geodesic distances.

### 5.2 Limitations

- **Sensitive to graph connectivity:** If the neighborhood graph is not well-chosen, the embedding can fail or break into separate components.
- **Computationally heavy:** All-pairs shortest paths and eigen-decomposition scale poorly with $N$.
- **Sensitive to noise and outliers:** Erroneous edges in the graph can distort geodesic distances.

## 6 Comparison with Other Methods

Table 1 compares Isomap with PCA, LLE, and t-SNE / UMAP. [2], [5], [6]

| Method | What it preserves | Strengths | Typical use |
|---|---|---|---|
| PCA | Global linear variance | Fast, simple, interpretable | Baseline linear reduction |
| Isomap | Geodesic distances on manifold | Captures global nonlinear geometry | Manifold structure analysis |
| LLE | Local linear reconstruction weights | Good local structure; manifold learning | Local geometry, visualization |
| t-SNE / UMAP | Local neighbor probabilities / fuzzy sets | Excellent cluster visualization | 2D/3D visualization of complex data |

Table 1: Isomap compared with PCA, LLE, and t-SNE/UMAP.

A practical rule of thumb:

- Use PCA for speed and baseline.
- Use Isomap when global manifold shape matters.
- Use LLE when preserving local linear relationships is key.
- Use t-SNE or UMAP for cluster visualization in 2D/3D. [5]

## 7 Applications

Isomap has been applied in various domains: [3], [4]

- **Swiss roll and synthetic manifolds:** Demonstrations of unfolding nonlinear structures.
- **Image manifolds:** Pose or lighting variations of objects, face images.
- **Speech and time series:** Discovering low-dimensional trajectories of dynamic systems.
- **Bioinformatics and neuroscience:** Understanding intrinsic structure in gene expression or neural activity.

## 8 Conclusion

Isomap provides a principled way to perform nonlinear dimensionality reduction by:

- Approximating geodesic distances on a neighborhood graph.
- Applying classical MDS to these distances to obtain a low-dimensional embedding.

It is especially suitable for data that lie on smooth manifolds where global geometry is important. While computational cost and sensitivity to graph construction are challenges, Isomap remains a foundational algorithm in manifold learning and a valuable tool for understanding complex high-dimensional data. [3], [15]

## References

[1] Wikipedia contributors, *Isomap*, https://en.wikipedia.org/wiki/Isomap, 2024.

[2] GeeksforGeeks, *Isomap - a non-linear dimensionality reduction technique*, https://www.geeksforgeeks.org/machine-learning/isomap-a-non-linear-dimensionality-reduction-technique/, 2024.

[3] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[4] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[5] Fiveable, *Dimensionality reduction methods*, https://fiveable.me/lists/dimensionality-reduction-methods, 2024.

[6] GeeksforGeeks, *Comparison of manifold learning methods in scikit learn*, https://www.geeksforgeeks.org/machine-learning/comparison-of-manifold-learning-methods-in-scikit-learn/, 2023.

[7] J. Chiquet, *Nonlinear dimensionality reduction methods (t-sne, umap, isomap, ...)* https://jchiquet.github.io/ds4m/tutorial_nonlinear_methods.html, 2020.

[8] X. Huang, Q. Huang, and Z. He, "Implementation of manifold learning algorithm isometric mapping," *Applied Mathematics*, vol. 10, no. 12, pp. 1187–1201, 2019, https://www.scirp.org/journal/paperinformation?paperid=96955.

[9] H. Li, *Manifold learning*, https://haifengl.github.io/manifold.html, 2019.

[10] GeeksforGeeks, *Isomap - a non-linear dimensionality reduction technique*, https://www.geeksforgeeks.org/machine-learning/isomap-a-non-linear-dimensionality-reduction-technique/, 2025.

[11] scikit-learn developers, *Isomap*, https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html, 2010.

[12] A. Elbling, *A visual introduction to dimensionality reduction with isomap*, https://alechelbling.com/blog/isomap/, 2022.

[13] M. Mkareshk, *Dimensionality reduction*, https://mkareshk.github.io/ml-interview/markdowns/Dimensionality%20Reduction.html, 2023.

[14] Z. Huo et al., "Low-rank isomap algorithm," *arXiv preprint arXiv:2103.04060*, 2021.

[15] M. W. Trosset and C. E. Priebe, "Rehabilitating isomap: Euclidean representation of geodesic distances," *arXiv preprint arXiv:2006.10858*, 2020.